NEC

PC-6001

いつでも どこでも

活用表

N60-BASIC

…になっています。

命　令		書　き　方
		説　　明
		実際の使い方

たとえば LIST という命令の場合,

LIST (命令)　　LIST〔開始行番号〕〔-〔終了行番号〕〕　(書き方)

プログラムを画面上に表示します　　(説　明)

LIST 30-80　　(実際の使い方)

は，行番号30番以上80番以下のプログラムを画面に表示します。

⋮

という様に書かれています。

※1．命令とパラメータとの間のスペースはどこでも無視されます。

PRINT　　　"ABC"

PRINT "ABC"

上の2つの文は同じことを意味します。

2．命令の書式は，書き方 に示すとおりです。

N60-BASIC

ABS

ABS (X)

ABS (X) はXの絶対値をとる命令です。

```
PRINT ABS(2-5)
```

を実行すると，　-3の絶対値 3が表示されます。

ASC

ASC (X$)

ASC (X$) は、文字列X$の最初の文字コードを与えます。

```
PRINT ASC("abc")
```

を実行すると，文字列abcのうちの最初の文字aのアスキーコード97を表示します。

CHR$

CHR$ (X)

CHR$ (X) はコードXに対応する文字を与える命令です。

```
PRINT CHR$(97)
```

を実行すると，コード97に対応する文字 a を表示します。

CLEAR

CLEAR[[m], n]

mは文字領域の大きさを，nはユーザメモリの上限を設定します。またすべての数値変数を 0，文字変数を空白にクリアします。

```
CLEAR 300, &hD000
```

を実行すると，文字領域の大きさが300，ユーザメモリの上限がD000(16進)番地に設定されます。(D000番地より上に，データや機械語プログラムを置くことができます。)

```
CLEAR 300
```

ユーザメモリの上限の設定が必要ないときは省略できます。

```
CLEAR
```

CLEARを実行すると，すべての数値変数を 0，文字変数を空白にクリアします。

CLOAD

CLOAD "ファイルネーム"

CLOADはカセットテープに記録されたプログラムを読み込む命令です。

```
CLOAD "ケイサン"
```

を実行すると，プログラムが記録されているカセットテープの中から　ケイサン　という名前（ファイルネーム）のついたプログラムをコンピュータ内部に読み込みます。

CLOAD

この場合，ファイルネームが省略されていますので，コンピュータは最初に見つけたプログラムを読み込みます。

CLOAD?

CLOAD?　ファイルネーム

コンピュータに記憶されているプログラムと，カセットテープに記憶されているプログラムを比較して等しければOKを，等しくなければBadを表示します。ファイルネームを省略すると，コンピュータに記憶されているプログラムとコンピュータが最初に見つけたプログラムを比較します。

CLS

CLS

を実行すると，画面全体がクリアされます。

COLOR

COLOR〔f〕〔, b〕〔, C〕

COLORは，この命令を実行したあとに画面に表示される，文字またはグラフィクスの色を決定するものです。
COLOR命令のパラメータf, b, cは，それぞれ次のことを表わします。

f：文字またはグラフィクスの色
b：背景の色
c：色の組合せ

```
10 COLOR 2
20 PRINT"ABC"
```

を実行してみて下さい。ABCは緑の反転した色で書かれます。次に

```
10 COLOR 3
20 PRINT "ABC"
```

を実行して下さい。今度は文字の色が橙色になります。このようにCOLOR命令の1番目のパラメータは，その後に書かれる文字の色を決定します。また線や点などの色も，同様に決定することができます(LINE, PSETの項を参照してください。)

```
10 SCREEN 2, 1, 1
20 COLOR ,3
30 CLS
```

このプログラムを実行すると画面全体が青色に変わります。

```
10 SCREEN 2, 1, 1
20 COLOR ,4
30 CLS
```

を実行すると，今度は画面全体の色が赤色になります。このようにCOLORの２番目のパラメータは背景の色を決定します。

```
 5 SCREEN 3, 2, 2
10 COLOR 2, 1, 1
20 PRINT "ABC"
30 GOTO 30
```

今度は３番目のパラメータです。上のプログラムを実行すると，緑のバックに黄色でABCを表示します。

```
 5 SCREEN 3, 2, 2
10 COLOR 2, 1, 2
20 PRINT "ABC"
30 GOTO 30
```

を実行すると，白のバックに水色でABCを表示します。このようにCOLORの３番目のパラメータは，画面全体の色の組合せを変える働きをします。

COLORのパラメータは画面のモード(SCREEN命令)により持つ意味が変わります。詳しくは26ページをご覧下さい。

CONSOLE

CONSOLE 〔l〕,〔m〕,〔n〕,〔o〕

CONSOLE 命令は，スクロール範囲の指定，ファンクションキー表示のON, OFFの指定，キーのクリック音のON, OFFの指定を行います。

l はスクロールの開始行，mはスクロールの長さ，nは１でファンクションキー表示ON, 0でOFF, Oは１でキークリック音ON, 0でOFFとなります。

```
CONSOLE 3,7
```

を実行すると，スクロールの範囲が３行目から，９行目になります。

```
CONSOLE ,,0
```

を実行すると，ファンクションキーの表示を消すことができます。

CONT

CONT

を実行すると，STOPキーまたはCTRL-Cキーによって停止したプログラムを停止した次の行から続けて実行します。

Xラジアンのコサインの値を計算します。

PRINT COS(3.14159265)

を実行すると，πラジアンのコサインの値 -1 を表示します。

CSAVE

CSAVE "ファイルネーム"

CSAVEはユーザーが作ったプログラムに名前を付けてカセットテープに記録する命令です。

CSAVE "ケイサン"

を実行すると，今作ったプログラムにケイサンという名前を付けて，カセットテープに書き込みます。

CSRLIN

CSRLIN

カーソルのY軸方向の位置を求める命令です。

```
10 LOCATE 10,5
20 PRINT CSRLIN
```

を実行すると，LOCATE命令で指定されたY座標5を表示します。

DATA

DATA 定数〔,定数〕……

DATA文は，READ文で読まれる数値や文字を割当てる働きをします。

DATA 1,2,A,B

DEFFN

DEFFN A (引数)＝関数の定義式

DEFFNはユーザーが作った関数を定義する命令です。

10 DEFFN A(X)＝X＊3

を実行すると，A(X)＝X＊3 というユーザーが作った関数が定義され，この後に

PRINT FN A(2)

を実行すると，Xに2を代入し，計算結果6を表示します。

なお，PC-6001では，引数は1種類しか使えません。

DIM

DIM A(n)

DIMは配列を宣言する命令で，A(0)～A(n)までの配列要素が確保されます。

DIM A(20)

を実行すると，Aという配列名で0番から20番までの配列要素が確保されます。

END END

ENDを実行すると，プログラムの実行を終了します。
STOPと違ってBreakを表示しません。

EXEC EXEC アドレス

EXECは機械語プログラムを実行する命令で，その先頭番地をアドレスに与えます。

```
EXEC &hD000
```

を実行すると，D000(16進)番地からの機械語プログラムを実行します。機械語プログラム実行後ベーシックに戻るには，機械語プログラムの最後にRET(C9)を置きます。またD000番地からの機械語プログラムはPOKE で書き込みます。

EXP EXP(X)

EXPは e^x を求める命令です。

```
PRINT EXP(3)
```

を実行すると，(e^3＝) 20.0855369を表示します。

FOR...TO...STEP NEXT FOR 変数名 ＝式 TO 式〔STEP 式〕

NEXTまでのプログラムを条件に従って繰返します。

```
10 FOR I=1 TO 10 STEP 2
20 PRINT I;
30 NEXT
```

ここで変数 I は，1 から 2 づつ10になるまで増え続けNEXTまでのプログラムを繰返し実行して，

```
1 3 5 7 9
```

を表示します。

FRE FRE(X)

FREはユーザメモリがどれほど残っているかを求める命令です。X はダミーです。

```
PRINT FRE(X)
```

を実行すると，ユーザメモリの残りを表示します。

```
PRINT FRE(X$)
```

括弧の中を文字型にすると，文字領域がどれほど残っているのかを表示します。

GOSUB 行番号

GOSUBは，指示された行番号から始まるサブルーチンをコールします。

GOSUB 500

を実行すると，行番号500から始まるサブルーチンを実行します。GOSUBはコールしたサブルーチンの最後に置かれるRETURN命令によって，GOSUBの次の行に戻ります。

GOTO

GOTO 行番号

GOTO は，指示された行番号から始まるプログラムにジャンプします。

GOTO 500

を実行すると,行番号500から始まるプログラムにジャンプします。

IF...THEN...

IF 式 THEN 文または行番号

条件式が真ならばTHENのあとの命令を実行します。

IF A=5 THEN 30

プログラムの流れの中で，Aの値が 5になったときだけ行番号30へジャンプして，そこからのプログラムを実行します。Aが 5以外のときは 次の行から実行を続けます。

INKEY$

INKEY$

INKEY$は，キーが押されたかどうかを調べる命令です。

```
10 A$= INKEY$
20 IF A$="" THEN 10
30 PRINT A$
```

上のプログラムを実行すると，キーが押されるまで待ち続け，押されたらそのキーを表示します。何も押さないときのINKEY$の値は "" で，ヌルストリングといいます。

INPUT

INPUT〔"プロンプト文";〕変数,〔変数…〕

INPUT文はキーボードからの入力を待ちます。

```
10 INPUT "つぎのデータは" ;N
20 PRINT N
```

を実行すると，画面に　つぎのデータは？と表示してからキーボードからの入力を待ちます。数値およびRETURNキーが押されると，PRINT文によって，その数が表示されます。

またディスプレイモードが3, 4のときにINPUT文を実 行すると，ディスプレイモード１の画面に戻ります。

INPUT#

INPUT#-入力機器番号，変数〔,変数〕

INPUT#-0：キーボード
INPUT#-1：カセットテープレコーダ
INPUT#-2：RS-232-C

INPUT#はINPUT文と似た命令ですが，後に指定される数値によって，入力機器を変えることができます。

```
10 INPUT#  -0, A
20 PRINT  A
```

を実行すると，指定された機器はキーボードですから，キーボードから数値およびRETURNキーを入力すると，画面にその値が表示されます。

```
10 INPUT#  -1, A$
20 PRINT  A$
```

この場合は入力機器にカセットが指定されています。実行すると，カセットテープから文字列を読み取って文字変数A$に代入し，表示します。ただし文字列データはあらかじめテープに記録しておかねばなりません。

```
10 INPUT#  -2, A$
20 PRINT  A$
```

を実行すると，RS-232Cインターフェイス(オプション)を介して，他のコンピュータやターミナルから文字列データを受取り表示します。

INP

INP (X)

I/OアドレスXのポートから読み込んだデータの値を求める命令です。

```
PRINT  INP (&h90)
```

INT

INT (X)

Xを超えない最大の整数を与えます。

```
PRINT  INT (2.4)
```

を実行すると2を表示します。

KEY

KEY ファンクションキーの番号，定義する文字列

ファンクションキーを定義します。

```
KEY1, "LINE"
```

を実行すると，F1にLINEが定義されます。また，ファンクションキーの表示は5文字までですが，8文字まで有効です。

COPYを実行すると，画面の情報をプリンターに出力します。プリンターは画面と同じ絵柄を出力します。

LEFT$

LEFT$(X$, X)

文字列 X$の中の左からX文字目までの文字列を求める命令です。

PRINT LEFT$("ABC", 2)

を実行すると，ABCの左から 2文字目までのABを表示します。

LEN

LEN(X$)

文字列 X$が何文字あるかを与えます。

PRINT LEN("ABCD")

を実行すると，4が表示されます。

LET

LET 変数名＝式

変数に文字または数値を割付けます。

LET A$＝"ABCD"

LET B＝35

LIST

LIST〔開始行番号〕〔ー〔終了行番号〕〕

プログラムを画面上に表示します。

LIST

を実行すると，プログラムのすべてを画面に表示します。

LIST 30-80

行番号30から80までのプログラムを画面に表示します。

LIST 30-

行番号30以上のすべてのプログラムを画面に表示します。

LIST -30

行番号30までのすべてのプログラムを画面に表示します。

LIST 30

行番号30のプログラムだけを画面に表示します。

LLIST

LLIST〔開始行番号〕〔ー〔終了行番号〕〕

プログラムリストをプリンターに出力します。

LLIST

LLIST 30-80

LLIST 30-

LLIST -30

LLIST 30

LINE

LINE (XI, Y1) - (X2, Y2) 〔,色〕〔,B〔F〕〕

2点 (X1, Y1), (X2, Y2) を指定した色の線で結びます。Bオプションでは，指定した2点を対角とする4角形を描きます。更にFオプションを加えると，4角形をぬりつぶします。

```
LINE (10, 10) - (90, 90) , 2
```

を実行すると，点 (10, 10) と (90, 90) を 2 で指定された色の線で結びます。

```
LINE (10, 10) - (90, 90) , 2, B
```

を実行すると，上のプログラムで描いた線を対角線とする4角形を描きます。

```
LINE (10, 10) - (90, 90) , 2, BF
```

を実行すると，前に描いた4角形をぬりつぶします。

LOCATE

LOCATE 水平位置，垂直位置

LOCATE は指定された位置にカーソルを移動する命令です。

```
10 LOCATE 10, 5
20 PRINT "ABC"
```

上のプログラムを実行すると,画面左上すみを (0, 0) として，(10, 5)の場所からABCを表示します。

LOG

LOG (X)

Xの自然対数を与えます。

```
PRINT LOG (2.71828183)
```

を実行すると，1を表示します。

LPOS

LPOS (X)

プリンターのヘッドの位置を与えます。Xはダミーです。

```
LPRINT "ABC"; LPOS (X)
```

を実行すると，プリンタにABCとヘッド位置 3 が出力されます。

LPRINT

LPRINT 式〔,式〕……

プリンタに対するプリント文。

```
LPRINT "ABC"
```

を実行すると，プリンタにはABCが出力されます。

MID$

MID$ (X$, X, Y)

文字列X$の左 X 番目から始まり，長さ Yの文字列を与
ます。

```
PRINT MID$ ("ABCDE", 2, 3)
```

を実行すると,文字列"ABCDE" の左 2番目からの 3 文
BCD を表示します。

NEW

NEW

メモリにあるプログラムと変数をすべてクリアします

ON...GOSUB

ON 式 GOSUB 行番号〔, 行番号〕…

式の値に対応して，相当するサブルーチンをコールし
す。

```
10 INPUT A
20 ON A GOSUB 40, 50, 60
30 GOTO 10
40 PRINT "A=1" : RETURN
50 PRINT "A=2" : RETURN
60 PRINT "A=3" : RETURN
```

上のプログラムを実行すると，入力する Aの値を 1, 2, 3
変えたときに，それぞれ行番号40, 50, 60, のサブルー
ンをコールします。

ON...GOTO

ON 式 GOTO 行番号〔,行番号〕…

式の値に対応して,相当する行番号に制御を移します。

```
10 INPUT A
20 ON A GOTO 40, 50, 60
30 GOTO 10
40 PRINT "A=1" : GOTO 10
50 PRINT "A=2" : GOTO 10
60 PRINT "A=3" : GOTO 10
```

上のプログラムを実行すると，入力する Aの値を1, 2,
と変えたときに，それぞれ行番号 40, 50, 60にジャン
します。

OUT

OUT ポート番号，データ

データを指定したポートに出力します。

```
OUT &hB0, &hF3
```

を実行するとポートB0にデータF3を出力します。

PAINT

PAINT (X, Y) 〔, 色〔, 境界色〕〕

PAINT命令は，(X, Y)で指定した位置から，指定した色で，指定された境界色までぬりつぶします。

```
10 SCREEN 3, 2, 2
20 LINE (10, 10) - (90, 90) , 2, B
30 PAINT (20, 20) , 3, 2
```

このプログラムを実行すると，2で指定された色で四角形を描いた後，3の色でぬりつぶします。四角形を描く色と，境界色が同じことに注意して下さい。

PEEK

PEEK(X)

メモリのX番地の内容を与えます。

```
PRINT PEEK (&hC000)
```

は，C000(16進)番地の内容を表示します。

PLAY

PLAY ボイスA文字列〔, ボイスB文字例〔, ボイスC文字例〕〕

PLAYは音楽を発生する命令で，最大3重奏が可能です。

```
PLAY "CDEFG"
```

を実行すると，ドレミファソを演奏します。

```
PLAY "C","E","G"
```

を実行するとドミソの和音が得られます。

PLAY命令の詳しい使い方は25ページの使用例をご覧下さい。

POINT

POINT (X, Y)

POINT命令は，座標(X, Y)で示される場所のCOLORの番号を与えます。

```
10 PSET (10, 10) , 3
20 PRINT POINT (10, 10)
```

を実行すると，行番号10で点(10, 10)を3の色に指定します。行番号20では同じ点(10, 10)の色の番号を読み取るので3を表示します。

POKE

POKE メモリ番地，データ

データメモリの指定したアドレスに対してデータを書込みます。

```
POKE &hC200, &h41
```

はC200(16進)番地に，データ41(16進)を書き込みます。

POS

POS (X)

POSはカーソルのX軸方向位置を求める命令です。

```
PRINT "ABC"; POS (X)
```

を実行すると，ABCと3を出力します。Xはダミーです。

PRESET

PRESET (X, Y)

を実行すると，画面に描かれていた座標(X, Y)のドットを消します。

```
10 PSET(10, 10), 3
20 FOR I=1 TO 300 : NEXT
30 PRESET(10, 10)
```

上のプログラムを実行すると，(10, 10)の場所を3の色に変えてから，再びもとの色に戻ります。

PRINT

PRINT〔,または；または空白,または式…〕

PRINTは画面に情報を出力する命令です。

```
10 A=5
20 PRINT "A="; A
```

を実行すると，A=5が表示されます。

PRINT #

PRINT #－出力機器番号，式〔,式〕…

PRINT#-0 ：ディスプレイ
PRINT#-1 ：カセットテープレコーダ＊
PRINT#-2 ：RS-232-C
PRINT#-3 ：プリンタ

PRINT #はPRINT文と似た命令ですが，指定する数値によって，出力機器を変えることができます。

```
PRINT#-0, "ABC"
```

を実行すると，画面にABCを表示します。

```
PRINT#-1, "ABC"
```

を実行すると，カセットテープレコーダに対してABCを出力します。

```
PRINT#-2, "ABC"
```

を実行すると，RS-232-Cインタフェース（オプション）を介して他のコンピュータやターミナルにABCを出力します。

```
PRINT#-3, "ABC"
```

を実行すると，プリンタに対してABCを出力します。

＊ PRINT #-1において複数の数値および文字列を記録する場合，次のようにします。

```
PRINT #−1, A$ "," B$"," C
```

PSET

PSET (X, Y) 〔,色〕

画面上の座標 (X, Y) の位置に指定した色でドットを表示します。

```
PSET(10, 10), 2
```

を実行すると，画面上の点 (10, 10) に指定した色2でドットを表示します。

READ

READ 変数〔,変数〕…

READ 文は, DATA 文のデータを各変数に読み込む命令です。

```
10 READ A, B
20 DATA 1, 2
30 PRINT A, B
```

を実行すると, 変数A, Bにはそれぞれ1, 2が割付けられ, 画面に表示されます。

REM

REM

REMはプログラムに注釈をつけるときに用います。
REMの行はマルチステートメント化はできません。

```
10 REM＊＊＊＊START＊＊＊＊
```

RESTORE

RESTORE 〔行番号〕

DATA文を, 指定した行番号から読めるようにします。

```
10 READ A, B
20 RESTORE
30 READ C, D
40 DATA 1, 2
50 DATA 3, 4
60 PRINT A, B, C, D
```

を実行すると, 変数A, BにDATA文のデータ1, 2を変数C, Dにもデータ1, 2を読み込んで画面に表示します。行番号20番のプログラムを,

```
20 RESTORE 50
```

と変更して実行すると, 変数A, Bにデータ1, 2, 変数C, Dにはデータ3, 4を読み込んで画面に表示します。このようにRESTOREの後に行番号を指定すると, その指定された行からデータを読みます。

RETURN

RETURN

サブルーチンから, そのサブルーチンをコールしたGOSUB文の次の文へ戻ります。

```
 10 GOSUB 100
 20 PRINT A
 30 END
100 A=2
110 RETURN
```

を実行すると, 行番号100番のサブルーチンをコールした後, GOSUB文の次の行(行番号20)に戻り, Aの値 2 を表示します。

RIGHT$

RIGHT$(X$, X)

文字列 X$ の右から X文字目までの文字列を求める命令です。

PRINT RIGHT$("ABCD", 2)

を実行すると，文字列ABCDのうちの右からの 2文字CDを表示します。

RND

RND(n)

0から 1の間の乱数を発生します。

n>0で通常の乱数を発生します。
n<0で乱数系列を更新します。
n=0で同じ乱数を繰返します。

```
10 PRINT RND(1)
20 GOTO 10
```

RUN

RUN〔行番号〕

RUNはプログラムの実行を開始させる命令です。

RUN 50

とすると，行番号50から実行を開始します。

SCREEN

SCREEN〔m〕〔, a〕〔, v〕

PC-6001 はメモリを拡張しない状態では，特に指定のないかぎり 2つの画面を持っています。これをそれぞれページ1。ページ 2と呼びます。電源を投入したときに表示されている画面がページ1 で，表示されていない画面がページ2です。どちらの画面にも文字やグラフィックスを描くことができ，表示させることができます。以上の操作はSCREEN命令の a と v のパラメータによって操作されます。もう 1つのパラメータ mは，ディスプレイモードを決定します。

m：ディスプレイモード
a：文字またはグラフィクスが有効になるページ
v：画面に表示されるページ

第1パラメータ　m（ディスプレイモード）

SCREEN 1	文字モード
SCREEN 2	64×48グラフィックスおよび文字
SCREEN 3	128×192グラフィックス
SCREEN 4	256×192グラフィックス

第2パラメータ a (文字またはグラフィクスが有効になるページ)

SCREEN □, 1	ページ1に文字またはグラフィクスを描く
SCREEN □, 2	ページ2 〃
SCREEN □, 3	ページ3 〃 (拡張RAMが必要)
SCREEN □, 4	ページ4 〃 (〃)

第3パラメータ v (画面に表示されるページ)

SCREEN □, □, 1	ページ1が画面に表示されます
SCREEN □, □, 2	ページ2 〃
SCREEN □, □, 3	ページ3 〃 (拡張RAMが必要)
SCREEN □, □, 4	ページ4 〃 (〃)

上の表を参照して，ディスプレイモード，文字またはグラフィクスを描くページ，画面に表示するページを決定して下さい。

```
10 SCREEN 2, 1, 1
20 CLS
30 LINE (0, 0) - (255, 191) , 7
```

(このプログラムを実行する前にリセットボタンを押して下さい。)

を実行すると，LINE命令で線を描いたページと，画面に表示されるページが両方とも1ですから，線が現われるのを見ることができます。

```
10 SCREEN 2, 1, 2
20 CLS
30 LINE (0, 0) - (255, 191) , 7
```

ところが上のプログラムでは，線を描いたのは1ページ，画面に表示されるのはページ2になりますから，線が現われるのを見ることができません。

ここで[⇅]キーを押すか，SCREEN, , 1 でページ1を表示画面に指定すると，線を見ることができます。

SCREEN (X,Y)

X, Yで指定した座標の文字のアスキーコードを与えます。

PRINT SCREEN (31, 0)

を実行すると画面右上すみの文字のアスキーコードを表示します。

SGN

SGN (X)

SGNは，Xの符号を判別する命令で，Xが負，ゼロ，正でそれぞれ，-1, 0, 1が与えられます。

PRINT SGN (3-5)

を実行すると，括弧の中は負の数となるので，-1が表示されます。

SIN

SIN (X)

Xラジアンのサインの値を計算します。

PRINT SIN (3.14159265/2)

を実行すると，π/2ラジアンのサインの値1を表示します。

SPC

SPC (n)

SPC (n) は，カーソルの後に n 個のスペースを与えます。

```
10 PRINT "ABC";
20 PRINT SPC (5);
30 PRINT "DEF"
```

を実行すると，ABCとDEFの間に 5 つのスペースが入ります。

SQR

SQR (X)

Xの平方根を求める命令です。

```
PRINT SQR (4)
```

を実行すると 4の平方根 2 を表示します。

STICK

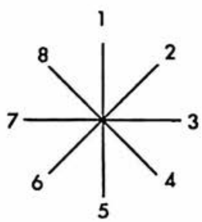
STICK (n)

ジョイスティックの状態を与えます。n はジョイスティックの番号を表わします。

n=0：キーボードカーソルキー
　1：ジョイスティック 1
　2：ジョイスティック 2

```
10 PRINT STICK (0)
20 GOTO 10
```

を実行すると，STICK (0) すなわちカーソルキーの状態を表示します。カーソルキーを押すと左図に示す値を表示示します。



カーソルキー(↑)を押すと 1，(→)を押すと 3，(↑)と(→)とを同時に押すと 2 をそれぞれ表示します。

ジョイスティック 1，2 においても，左図のようにレバーを倒せば，対応した数値が与えられます。

```
10 PRINT STICK (0);
20 PRINT STICK (1);
30 PRINT STICK (2)
40 GOTO 10
```

上のプログラムですべてのジョイスティックの状態を調べることができます。

STOP

STOP

を実行すると，プログラムの実行を停止し，Break メッセージを出力してコマンド待ちに戻ります。

STRIG

STRIG (n)

ジョイスティックのトリガーボタンの状態を与えます。
n はジョイスティックの番号を表わします。

n=0：キーボードスペースキー
　1：ジョイスティック 1 ボタン
　2：ジョイスティック 2 ボタン

```
10 PRINT STRIG (0)
20 GOTO 10
```

を実行すると，キーボードのスペースキーが押されていれば 1，はなされていれば 0 を表示します。

STR$

STR$ (X)

数値Xを，それを表現する文字列に変換します。

```
PRINT STR$ (35) +"円"
```

を実行すると，35円を表示します。

SOUND

SOUND レジスタ番号，データ

SOUNDはPC-6001本体内に実装されているサウンドICに直接呼びかける命令で，1 番目のパラメータはIC内部のレジスタの番号，2 番目のパラメータはレジスタに入れるデータを表わします。

```
10 SOUND 7, 254
20 SOUND 8, 8
30 SOUND 0, 255
40 FOR I=1 TO 900 : NEXT
50 SOUND 8, 0
```

このプログラムを実行すると，一定時間音を発生します。それぞれのレジスタがどんな意味を持っているかは巻末の表および計算例をご覧下さい。

TAB

TAB (n)

TABは画面左端より n 個のスペースを与えます。

```
10 PRINT "ABC";
20 PRINT TAB (5);
30 PRINT "DEF"
```

このプログラムを実行すると，ABCとDEFの間には 2 つのスペースが入ります。

TAN

TAN (X)

Xラジアンのタンジェントを与えます。

```
PRINT TAN (0)
```

を実行すると 0 ラジアンのタンジェントの値 0 を表示します。

TIME

TIME

(1/512) 秒ごとに最下位の桁が 2 ずつ増えるタイマーを与えます。ですから 4 桁目が，約 1 秒 (0.976525秒) になります。

これを補正して 1 秒を単位とするタイマーのプログラムを示します。

```
10 CLS
20 X=TIME
30 Y=TIME-X
40 Z=INT(Y/(1000/1.024))
50 LOCATE 15,8 : PRINT Z
60 GOTO 30
```

なおTIME命令はハードウエアの影響を受けるため，正確なタイマーとしては使用できません。

USR

USR(X)

USRは，ユーザーが作った機械語プログラムを実行させる命令です。X は機械語プログラムに受け渡す値を示します。USR関数を使うとき，ユーザーの機械語プログラムが何番地から始まっているのか，PC-6001に知らせる必要があります。これは，FAEB(16進)番地と，FAEC(16進)番地に，それぞれ，機械語プログラム先頭アドレスの，下位 8 ビット，上位 8 ビットをPOKE文で書き込むことによって行います。

次にベーシックプログラムから機械語プログラムにデータ X を受け渡すには，機械語プログラムで，741(16進)番地をコールします。CPUのDEレジスタに，データ X が代入されます。X の範囲は-32768から32677の間です。

機械語プログラムから，ベーシックプログラムにデータを受け渡すには，A, Bレジスタをペアレジスタとして使って，データを入れた後に，D16(16進) にジャンプさせると，USR関数の値が，ABレジスタの値となります。

では，データ X より 1 大きい数を機械語で求めて，ベーシックに受け渡すプログラムを紹介します。

```
10 CLEAR 50,&hCFFF
20 ADDR=&hD000
30 FOR I=1 TO 9
40 READ A$
50 A=VAL("&h"+A$)
60 POKE ADDR,A
70 ADDR=ADDR+1
80 NEXT
```

```
 90 POKE &hFAEB,00 : POKE &hFAEC,&hD0
100 INPUT "データ"; X
110 ANS=USR(X)
120 PRINT ANS
130 GOTO 100
140 DATA CD,41,07,13,7A,43,C3,16,0D
```

VAL　　VAL (X$)

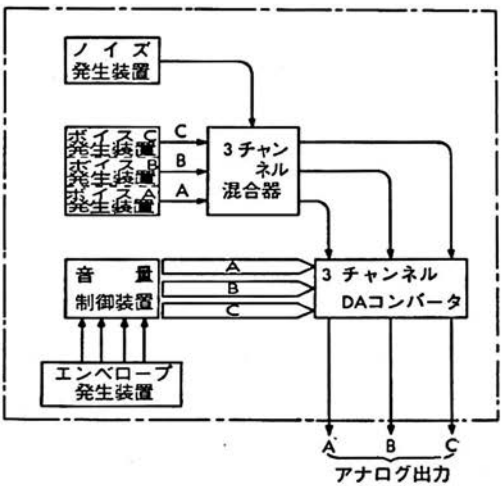
数字（文字列X$）を数値に変換します。

```
10 A$="3E"
20 PRINT VAL("&h"+A$)
```

を実行すると，文字列　&h3E　を数値62（10進）に変換して表示します。

SOUND命令を行うために

（デモテープのPC-6001ミュージックおよびサウンド機能説明を参考にして下さい。）



PC-6001本体内に実装されているSOUND ICの内部は，上図のようになっています。

音の発生装置と音量の制御装置は3チャンネルずつ分離されているので，まったく独立にコントロールすることができます。しかし，ノイズ発生装置と，エンベロープ発生装置は1つしかありませんので，各チャンネルで共通に使うことになります。

図に示したこれらの装置をコントロールするために，SOUND ICの中には次ページのように16個のレジスタが用意されています。

SOUND レジスタ表

レジスタ＼ビット		B7	B6	B5	B4	B3	B2	B1	B0
R 0	チャンネルA周波数	8BIT FT A							
R 1						4BIT CT A			
R 2	チャンネルB周波数	8BIT FT B							
R 3						4BIT CT B			
R 4	チャンネルC周波数	8BIT FT C							
R 5						4BIT CT C			
R 6	ノイズ周波数				5BIT NP				
R 7	チャンネル設定	$\overline{\mathrm{IN}}$/OUT		$\overline{\mathrm{NOISE}}$			$\overline{\mathrm{TONE}}$		
		IOB	IOA	C	B	A	C	B	A
R 8	チャンネルA音量				M	L3	L2	L1	L0
R 9	チャンネルB音量				M	L3	L2	L1	L0
R10	チャンネルC音量				M	L3	L2	L1	L0
R11	エンベロープ周期	8BIT FT							
R12		8BIT CT							
R13	エンベロープ形状					E3	E2	E1	E0
R14	I/OポートAデータ	8BIT パラレル I/OポートA							
R15	I/OポートBデータ	8BIT パラレル I/OポートB							

これらのレジスタについて簡単に説明します。

〔R0, R1〕,〔R2, R3〕,〔R4, R5〕は，それぞれチャンネルA, B, Cの周波数を決定します。周波数の決定には，2つのレジスタを使います。

〔R6〕は，ノイズ周波数を決定します。

〔R7〕は，どのチャンネルを動作させるかを決定するレジスタで，A,B,Cすべてのチャンネルから音を出すことも，まったく音を出さないこともできます。またチャンネルA, B, Cのいずれにもノイズを混ぜることができます。

〔R8〕,〔R9〕,〔R10〕は，それぞれチャンネルA, B, Cの音量を決定するレジスタで，L0からL3の4ビットに0を入れると無音に，15を入れると最大の音量になります。また,〔R8〕,〔R9〕,〔R10〕の4ビット目に1を設定すると音の大きさが時間的に変化するエンベロープモードになります。エンベロープモードを指定した場合は，その音量の変化する周期，形状などを更に設定せねばなりません。

〔R11〕,〔R12〕はエンベロープの周期を決定します。周期の範囲が広いので，2つのレジスタで表現します。

〔R13〕は，エンベロープの形状を指定するレジスタです。

実際のデータの決め方は，以下を参考にして下さい。

SOUND レジスタ設定値について。

○周波数設定のしかた。

周波数設定レジスタR0, R1 (チャンネルA) に入れる値は次式によって計算できます。

$$TP_{10} = \frac{fclock}{16 \times fT} \quad , \quad CT_{10} + \frac{FT_{10}}{256} = \frac{TP_{10}}{256}$$

ここで

fclock	：	1.99675MHz
fT	：	出力する周波数
FT_{10}	：	FTA (R0) の値 (10進)
CT_{10}	：	CTA (R1) の値 (10進)

今，出力したい周波数を1KHzとすると，

$$TP_{10} = \frac{1.99675 \times 10^6}{16 \times (1 \times 10^3)} \doteqdot 125$$

$$CT_{10} + \frac{FT_{10}}{256} = \frac{125}{256}$$

∴　R0にはFT_{10} =125 (10進)

　　R1にはCT_{10} =0

を設定すればよい。

○ノイズ周波数設定のしかた

ノイズ周波数設定レジスタR6に入れる値は次式によって計算できます。

$$NP_{10} = \frac{fclock}{16 \times fN}$$

ここで

fclockは		1.99675〔MHz〕
fN	は	出力したいノイズ周波数〔Hz〕
NP_{10}	は	NP (R6) に設定する値 (10進)

今，出力したいノイズ周波数を20KHzとすると，

$$NP_{10} = \frac{1.99675 \times 10^6}{16 \times (20 \times 10^3)} \doteqdot 6$$

∴R6にはNP_{10} =6 (10進) を設定すればよい。

○チャンネルの設定

チャンネル設定レジスタにおいて，使用したいチャンネルのビットのみを0にします。

	B7	B6	B5	B4	B3	B2	B1	B0
R7	$\overline{\text{IN}}$/OUT		$\overline{\text{NOISE}}$			$\overline{\text{TONE}}$		
	IOB	IOA	C	B	A	C	B	A

たとえば，チャンネルAのTONEとチャンネルAのNOISEのみを使いたい場合は，

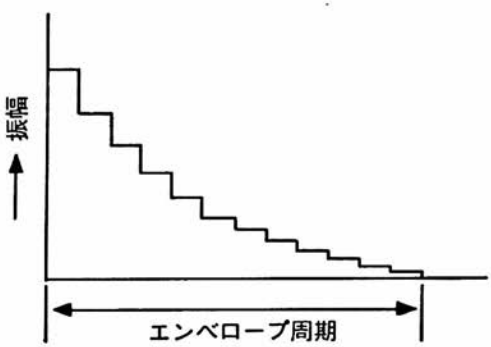
1　1　1　1　0　1　1　0

をR7に設定します。

○音量設定のしかた

音量設定では，大きさが一定のモードと，変化するモード(エンベロープモード)を選ぶことができます。音量が一定のときは，R8(チャンネルAの場合)の下位4ビットに0から15(10進)を入れることによって設定され，0のときに無音になります。

音量が時間的に変化するエンベロープモードはR8の4ビット目を1にすることにより設定できます。この場合どのような形状で振幅を増加または減衰させるのか，またその周期がどれほどなのかを更に指定せねばなりません。



○エンベロープの周期は次式で計算できます。

$$EP_{10}=\frac{fclock}{256fE}\quad,\qquad CT_{10}+\frac{FT_{10}}{256}=\frac{EP_{10}}{256}$$

ここで

fclock	：	1.99675MHz
fE	：	音が増加または減衰する周波数Hz
FT_{10}	：	FT (R11) に設定する値 (10進)
CT_{10}	：	CT (R12) に設定する値 (10進)

今, 音が減衰してゼロになるまでの時間を2秒に設定しようとすると,
$fE = 1/2 = 0.5Hz$

$$EP_{10} = \frac{1.99675 \times 10^6}{256 \times 0.5} = 15,600$$

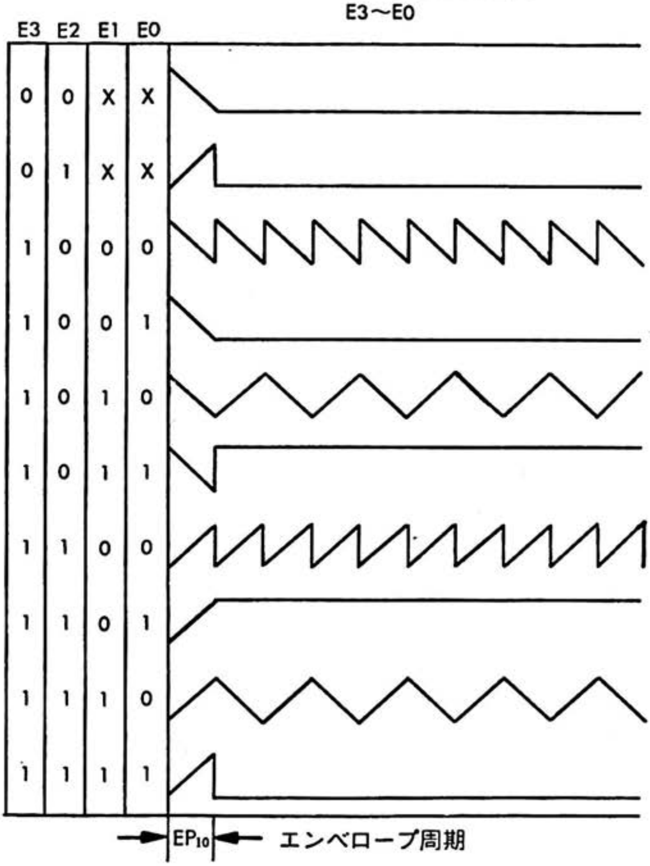
$$CT_{10} + \frac{FT_{10}}{256} = \frac{15600}{256} = 60 + \frac{240}{256}$$

∴R11にはFT_{10} =240 (10進)
　R12にはCT_{10} =60 (10進) を設定すればよい。

○エンベロープ形状の選択

　エンベロープ形状レジスタR13のE0からE3の値より形状を設定することができます。
E0からE3の値がどのような波形に対応しているのかを下図に示します。
なお, 図中のXは意味をもちません。(0でも1でもかまわない)

エンベロープ発生器出力
E3〜E0



PLAY命令の使い方 (デモテープのPC-6001ミュージックおよびサウンド機能説明を参考にして下さい。)

PLAY文の中で次の文字は特別な意味を持ちます。

M X　エンベロープ周期の設定 (1≦ X ≦65535)
S X　エンベロープ形状の設定 (0≦ X ≦15)
V X　音量の設定 (0≦ X ≦15)
L X　長さの設定 (1≦ X ≦64)
O X　オクターブの設定 (1≦ X ≦8)
T X　テンポの設定 (3≦ X ≦255)
R X　休符の設定 (1≦ X ≦64)
X ＋　音を半音上げます。
X ー　音を半音下げます。

PLAY

1. エンベロープを設定して音を発生させます。
"S13M300CDE"
2. 伴奏を付けます。
"S13M300CDE", "EEG"
3. オクターブを変えます。
"S13M300O5CDE", "EEG"
4. テンポを変えます。(早くします)
"T200S13M300O5CDE","1200EEG"
5. 伴奏を大きな音にします。
"T200S13M300O5CDE","T200V12EEG"
6. 休符を入れます。
"T200S13M300O5CR8DR8E","T200V12ER8ER8G"
7. 音の長さを変えます。
"T200S13M300O5L2CR8DR8E","T200V12L2ER8ER8G"
8. 半音上げます。
"T200S13M300O5L2C+R8DR8E","T200V12L2ER8ER8G"
9. 数で音階を表わします。

```
10 FOR I=0 TO 96
20 PLAY "n=I;"
30 NEXT
```

COLOR文 （デモテープのPC-6001 画面およびグラフィクス機能説明を参考にして下さい。）

COLOR〔f〕〔, b〕〔, C〕

f, b, cはそれぞれ次のことを表わします。

f：文字またはグラフィクスの色

b：背景の色

c：色の組合せ

画面のモード(SCREEN命令の最初のパラメータ)によって，同じ数字でも表示される色が異なりますので，画面のモード別にCOLOR文の説明をします。

1. 文字モード(SCREEN1)におけるCOLOR文

f	b	c	文字またはグラフィックスの色
1	X	1	緑
2	X	1	緑　反　転
3	X	1	橙
4	X	1	橙　反　転
1	X	2	橙
2	X	2	橙　反　転
3	X	2	緑
4	X	2	緑　反　転

文字モードの画面でCOLOR文を実行した後にCLSを実行することによって画面全体の色を文字の色(fの色)にすることができます。

(表中のXは意味を持ちません。)

2. 64×48グラフィクスモードにおけるCOLOR文

COLOR文はその後に画面に表示される文字またはグラフィクスに対して有効となる命令です。

f	b	c	文字の色	グラフィクスの色
0	0〜8	1	緑	黒
1	0〜8	1	緑	緑
2	0〜8	1	緑 反 転	黄
3	0〜8	1	橙	青
4	0〜8	1	橙 反 転	赤
5	0〜8	1	橙 反 転	白
6	0〜8	1	橙 反 転	水　色
7	0〜8	1	橙 反 転	紫
8	0〜8	1	橙 反 転	橙
0	0〜8	2	橙	黒
1	0〜8	2	橙	白
2	0〜8	2	橙 反 転	水　色
3	0〜8	2	緑	紫
4	0〜8	2	緑 反 転	橙
5	0〜8	2	緑 反 転	緑
6	0〜8	2	緑 反 転	黄
7	0〜8	2	緑 反 転	青
8	0〜8	2	緑 反 転	赤

64×48グラフィクスモードでは，グラフィクスの色はfとCによって決定され，9色が使用できます。しかし文字の色は文字モードのときと同じ緑，緑反転，橙，橙反転の4色です。fが5以上に指定されたときの文字の色はすべてf=4の色となります。

画面全体の色を変えるには，bに0～8(色の割当ては，グラフィクスの色と同じ)を設定してCOLOR文を実行し，その後にCLSを実行することで行えます。

3. 128×192グラフィクスモードにおけるCOLOR文

COLOR文はその後に画面に表示される文字またはグラフィクスに対して有効となる命令です。

f	b	c	文字またはグラフィクスの色
1	1～4	1	緑
2	1～4	1	黄
3	1～4	1	青
4	1～4	1	赤
1	1～4	2	白
2	1～4	2	水色
3	1～4	2	紫
4	1～4	2	橙

128×192グラフィクスモードで画面全体の色を変えるときは，bに1～4を設定してCOLOR文を実行した後，CLSを実行することにより実現できます。

4. 256×192グラフィクスモードにおけるCOLOR文

f	b	c	文字またはグラフィクスの色
1	0	1	黒地に緑
0	1	1	緑地に黒
1	0	2	黒地に白
0	1	2	白地に黒

256×192グラフィクスモードで画面全体の色を変えるには，bに0～1を設定してCOLOR文を実行した後，CLSを実行することにより実現できます。

特 殊 文 字

コントロールキーの役割

CTRL B　カーソルを１項目ごとに左へずらす。
CTRL C　プログラムの実行を中止して，コマンド待ちに戻る。
CTRL E　カーソルキーから後をまっ消する
CTRL F　カーソルを１項目ごとに右へずらす。
CTRL G　ブザーを鳴らす。
CTRL H　１文字削除
CTRL I　TAB。8カラムごとにカーソルを右へ移動する。
CTRL J　ラインフィード
CTRL K　ホーム。カーソルを左上に戻す。
CTRL L　画面全体のクリア
CTRL M　キャリッジ リターン
CTRL R　INSと同じ
CTRL U　カーソルのある行を１行まっ消する。
&H　16進定数を表わす。
？(疑問符) PRINTと同じ(ただしL？はLPRINTとして使えない)
:(コロン)　マルチステートメントの場合の区切り記号。
;(セミコロン) PRINT文中での区切り記号。

エラーメッセージ一覧

?BS Error(Bad Subscript Error)
配列の添字が宣言した大きさを超えています。
配列の添字の個数が宣言したときと違っています。

?CN Error(Continue Error)
CONT命令は，プログラム中に誤まりがあったり，プログラムリストを変更した後には実行できません。

?DD Error(Dupulicate Definition Error)
すでに宣言された配列と同じ名前の配列をDIM命令で再び宣言しています。

?Extra Ignored
INPUT文で指定した変数の数より，多いデータを入力しています。

?FC Error(Function Call Error)
許されている範囲外の数を使っています。

?FD Error(File Data Error)
データの構成が不適当です。

?ID Error(Illegal Direct Error)
ダイレクトモードでINPUT文，DEFFN文が使用されています。

?LS Error(Long String Error)
文字列の長さが255を超えています。

?MO Error(Missing Operand Error)
代入命令 "＝" の右辺がないなど，記述が不完全です。



?NF Error(Next without For Error)
FORがないのにNEXTが使用されています。

?OD Error(Out of Data Error)
READ命令を実行しようとしても，DATAがありません。

?OM Error(Out of Memory Error)
プログラムが大きすぎてメモリが足りません。
サブルーチンが自分自身をコールしています。

?OS Error(Out of String Space Error)
文字列用の領域が足りません。CLEAR文によって増やして下さい。

?OV Error(Over flow Error)
計算結果が大きすぎ，または小さすぎます。

?RG Error(Return without Gosub Error)
GOSUB命令がないのにRETURNが使われています。

?SN Error(Syntax Error)
文字を書き違っています。

?ST Error(String too Complex Error)
文字列の計算が複雑すぎます。

?TM Error(Type Mismatch Error)
文字と数値とをいっしょに取扱っています。

?TR Error(Tape Read Error)
テープの読み込みが正しく行なわれていません。

?UF Error(Undefined Function Error)
DEF FNで定義していないのに〝FN″で始まる関数を使っています。

?UL Error(Undefined Line number Error)
GOTO文，GOSUB文，RESTORE文，RUN命令で指定した行番号がありません。
プログラム訂正時に存在しない行を消そうとしています。

?/0 Error(Division by Zero Error)
0で割算をしています。
TAN(X)を計算するときに，引数Xが$\pi/2$(1.57079633)になっています。

演 算 子

記号	機能
＝	代入または等値判定
−	負数表示または減算
＋	加算または文字列の結合
＊	乗算
/	除算
∧	べき算
NOT	否定または2の補数(結果は整数)
AND	論理積(結果は整数)
OR	論理和(結果は整数)
=, <, >, <=, =<, >=, =>, <>, ><	条件判定(整数，真ならば-1,偽ならば0)

演算子の優先順位

(1) 括弧でくくられた式
(2) べき乗　(A∧B)
(3) 負数　(- A)
(4) ＊，/
(5) ＋，−
(6) 条件判定
(7) NOT
(8) AND
(9) OR

■キャラクタコード表

下位4ビット ＼ 上位4ビット	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		x		0	@	P		p	♠			ー	タ	ミ	た	み
1	月	┴	!	1	A	Q	a	q	♥	あ	。	ア	チ	ム	ち	む
2	火	┬	"	2	B	R	b	r	♣	い	「	イ	ツ	メ	つ	め
3	水	┤	#	3	C	S	c	s	◆	う	」	ウ	テ	モ	て	も
4	木	├	$	4	D	T	d	t	○	え	、	エ	ト	ヤ	と	や
5	金	┼	%	5	E	U	e	u	●	お	・	オ	ナ	ユ	な	ゆ
6	土	│	&	6	F	V	f	v	を	か	ヲ	カ	ニ	ヨ	に	よ
7	日	─	'	7	G	W	g	w	ぁ	き	ァ	キ	ヌ	ラ	ぬ	ら
8	年	┌	(	8	H	X	h	x	ぃ	く	ィ	ク	ネ	リ	ね	り
9	円	┐	)	9	I	Y	i	y	ぅ	け	ゥ	ケ	ノ	ル	の	る
A	時	└	*	:	J	Z	j	z	ぇ	こ	ェ	コ	ハ	レ	は	れ
B	分	┘	+	;	K	[	k	{	ぉ	さ	ォ	サ	ヒ	ロ	ひ	ろ
C	秒	×	,	<	L	¥	l	¦	ゃ	し	ャ	シ	フ	ワ	ふ	わ
D	百	大	−	=	M	]	m	}	ゅ	す	ュ	ス	ヘ	ン	へ	ん
E	千	中	.	>	N	^	n	~	ょ	せ	ョ	セ	ホ	゛	ほ	
F	万	小	/	?	O	_	o		っ	そ	ッ	ソ	マ	゜	ま	

NEC日本電気株式会社 ・ 新日本電気株式会社

パーソナルコンピュータ事業部パーソナルコンピュータ販売部
〒108 東京都港区芝5丁目33-7（徳栄ビル）
電話 (03)453-5511(大代)

パーソナルコンピュータ・ディスプレイ事業部 販売部
〒213 川崎市高津区久本210番地
電話 (044)833-5201(大代)

PC-6101
78110412